

SPS-Programmierung mit dem Raspberry Pi und dem OpenPLC-Projekt

ModbusRTU- und ModbusTCP-Beispiele
mit dem Arduino Uno und ESP8266

#	Name	Klasse	Typ
1	Ein	Lokal	BOOL
2	Aus	Lokal	BOOL
3	LED	Lokal	BOOL
4	LED_Modbus	Lokal	BOOL
5	LEDblink	Lokal	BOOL
6	LEDTMP	Lokal	BOOL
7	BTN_HMI1	Lokal	BOOL

Josef Bernhardt

SPS-Programmierung mit dem Raspberry Pi und dem OpenPLC-Projekt

Einführung in die SPS-Programmierung mit dem Open-Source-Projekt

ModbusRTU- und ModbusTCP-Beispiele mit dem Arduino Uno und ESP8266



Josef Bernhardt

● © 2021: Elektor Verlag GmbH, Aachen.

1. Auflage 2021

● Alle Rechte vorbehalten.

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch auszugsweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet.

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen können auch dann eingetragene Warenzeichen sein, wenn darauf nicht besonders hingewiesen wird. Sie gehören dem jeweiligen Warenzeicheninhaber und unterliegen gesetzlichen Bestimmungen.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für die Mitteilung eventueller Fehler sind Verlag und Autor dankbar.

● Erklärung

Der Autor und der Herausgeber dieses Buches haben alle Anstrengungen unternommen, um die Richtigkeit der in diesem Buch enthaltenen Informationen sicherzustellen. Sie übernehmen keine Haftung für Verluste oder Schäden, die durch Fehler oder Auslassungen in diesem Buch verursacht werden, unabhängig davon, ob diese Fehler oder Auslassungen auf Fahrlässigkeit, Unfall oder andere Ursachen zurückzuführen sind.

Umschlaggestaltung: Elektor, Aachen

Satz und Aufmachung: D-Vision, Julian van den Berg | Oss (NL)

Druck: Ipskamp Printing, Enschede, Niederlande

● ISBN 978-3-89576-439-4

Ebook 978-3-89576-440-0

Elektor-Verlag GmbH, Aachen

www.elektor.de

Elektor ist Teil der Unternehmensgruppe Elektor International Media (EIM), der weltweit wichtigsten Quelle für technische Informationen und Elektronik-Produkte für Ingenieure und Elektronik-Entwickler und für Firmen, die diese Fachleute beschäftigen. Das internationale Team von Elektor entwickelt Tag für Tag hochwertige Inhalte für Entwickler und DIY-Elektroniker, die über verschiedene Medien (Magazine, Videos, digitale Medien sowie Social Media) in zahlreichen Sprachen verbreitet werden. www.elektor.de

Vorwort	7
Einführung	8
Kapitel 1 • Installation des Raspberry Pi 4	11
1.1 Hardwarebeschreibung	11
1.2 Installation des Betriebssystems	12
1.3 Installation des VNC Viewers	15
1.4 Installation der File Transfer Software WinSCP	18
1.5 Installation der openplcproject-Runtime	22
Kapitel 2 • Installation des OpenPLC-Editors	32
2.1 Download und Installation	32
2.2 Pin-Beschreibung des Raspberry Pi	37
2.3 Zusätzliche Hardware E/A Testboard	39
2.4 Zusätzliche Hardware 24V-SPS-Board	40
Kapitel 3 • Der OpenPLC-Editor	43
3.1 Beschreibung des OpenPLC-Editors	43
3.2 Kontaktplan Beispiel (KOP)	53
3.3 Funktionsbaustein Beispiel (FUP, FBD)	63
3.4 Anweisungsliste Beispiel (AWL)	71
3.5 Strukturierter-Text-Beispiele (ST, SCL)	77
3.5.1 Variable	77
3.5.2 Kontrollstrukturen	79
3.5.3 Konvertierungsoperatoren	83
3.5.4 Standardfunktionsbausteine nach IEC 61131-3	84
3.5.5 Erstes Programmbeispiel ST	86
3.5.6 ST-Beispiel zum Ansteuern eines Förderbandes	88
3.5.7 Definieren von Arrays mit dem OpenPLC-Editor	91
3.5.8 Definieren von Strukturen mit dem OpenPLC-Editor	95
3.5.9 Kombination von Strukturen mit Arrays mit dem OpenPLC-Editor	98
3.5.10 Definition von ENUMs	100
3.6 Ablaufsprache-Beispiel (AS)	104
Kapitel 4 • OpenPLC und der Modbus	113
4.1 Testen von SPS-Programmen mit Modbus-TCP	113

4.2 Visualisieren von SPS-Programmen mit AdvancedHMI	122
4.3 Visualisieren von SPS-Programmen übers Internet	131
Kapitel 5 • Modbus E/A Module	138
5.1 Modbus RTU Modul mit dem Arduino Uno	138
5.2 Modbus-TCP Modul mit dem ESP8266 und WLAN.	150
5.3 Webserver Anwendung mit dem ESP8266-E/A-Modul.	159
Kapitel 6 • Anhang	165
6.1 Literaturverzeichnis	165
6.2 Weblinks.	166
6.3 Modbus-Kommandos für das ESP8266-E/A-Modul	168
6.4 Schaltpläne und Layouts.	172
6.4.1 Testboard mit Tastern und LEDs:	172
6.4.2 Testboard mit 24V Ein-Ausgängen:	174
6.4.3 ESP8266-Board mit 24V Ein-Ausgängen:	176
Stichwortverzeichnis	182

Vorwort

Dieses Buch soll dem Leser eine praxisnahe Einführung bieten, um den Raspberry Pi als SPS für eigene Projekte einsetzen zu können.

Das Projekt wurde durch die Programmierer Edouard Tisserant und Mario de Sousa möglich. Diese haben nach der Einführung der IEC Norm 61131-3 im Jahr 2003 das „Matic-Projekt“ begonnen. Damit war es möglich, die in der Norm eingeführten Programmiersprachen in C-Programme zu übersetzen.

Später, als der Raspberry Pi immer populärer wurde, begann Thiago Alves mit dem „openplproject“. Er erweiterte den Editor aus dem „Beremiz-Projekt“ und schrieb eine Laufzeitbibliothek sowie eine Weboberfläche für den Raspberry Pi und den PC.

Jetzt wurde es möglich, Programme auf dem PC zu schreiben und diese auf den Raspberry Pi zu installieren.

Viele Benutzer des Raspberry Pi haben jetzt die Möglichkeit, eigene Steuerungen und Regelungen auf ihrer Hardware zu verwirklichen. Auch für Schulungszwecke ist die Hard- und Software wegen der Anlehnung an die IEC-Norm hervorragend geeignet.

Auch Einsteiger und Profis erfahren hier alles über die Installation und die Programmierung in den fünf Programmiersprachen, um eigene Steuerungen aufzubauen.

In einem späteren Kapitel wird auf die Visualisierung mit AdvancedHMI eingegangen, um Prozesse auf dem Bildschirm darzustellen.

Schaltungen mit Arduino und ESP8266, die für Modbus nötig sind, werden ebenfalls erläutert.

Wir wünschen dem Leser viel Erfolg bei der Lektüre des Buches.

Bad Abbach, Januar 2021
Josef Bernhardt

Einführung

Speicherprogrammierbare Steuerungen (SPS) haben die industrielle Steuerungstechnik revolutioniert. SPS werden seit ihrer Erfindung durch Richard E. Morley vor ca. 50 Jahren überwiegend in Industriesteuerungen und in der Hausautomation eingesetzt.

Hier die Definition der SPS nach EN61131-1

„Eine SPS ist ein digital arbeitendes elektronisches System für den Einsatz in industriellen Umgebungen mit einem programmierbaren Speicher zur internen Speicherung der anwenderorientierten Steuerungsanweisungen zur Implementierung spezifischer Funktionen wie z.B. Verknüpfungssteuerung, Ablaufsteuerung, Zeit-, Zähl- und arithmetische Funktionen, um durch digitale oder analoge Eingangs- und Ausgangssignale verschiedene Arten von Maschinen und Prozessen zu steuern.“

Der Raspberry Pi eignet sich wegen seiner Architektur mit der GPIO-Steckerleiste und auch wegen des niedrigen Preises hervorragend für eine Anwendung als SPS.

Auf dem Markt sind verschiedene fertige SPS auf Basis des Raspberry Pi verfügbar.

Ein großer Vorteil der SPS-Programmierung ist, dass sich der Programmierer nicht in Hardwaredetails der E/A Leitungen einarbeiten muss. Die analogen und Digitalen Aus- und Eingänge werden wie Variable behandelt. Dies ist auch mit Modulen möglich, die über ein Netzwerk mit einem Protokoll wie z. B. Modbus-TCP mit dem Raspberry Pi verbunden sind.

Ein weiterer Vorteil der SPS-Programmierung ist die Kompatibilität der SPS-Systeme untereinander. Programme für eine SPS vom Hersteller A können meist ohne großen Aufwand für die SPS vom Hersteller B verwendet werden.



Abbildung 0.1 Raspberry Pi SPS Unipi 1.1

In der Abb. 0.1 sehen wir eine SPS mit 24V-Eingängen und Relais Ausgängen und angebautem Raspberry Pi.

Die meisten SPS-Systeme unterstützen grafische und textuelle Programmiersprachen. Das openplcproject unterstützt vollständig die Norm IEC 61131-3, welche die grundlegende Softwarearchitektur und die Programmiersprachen für SPS definiert.

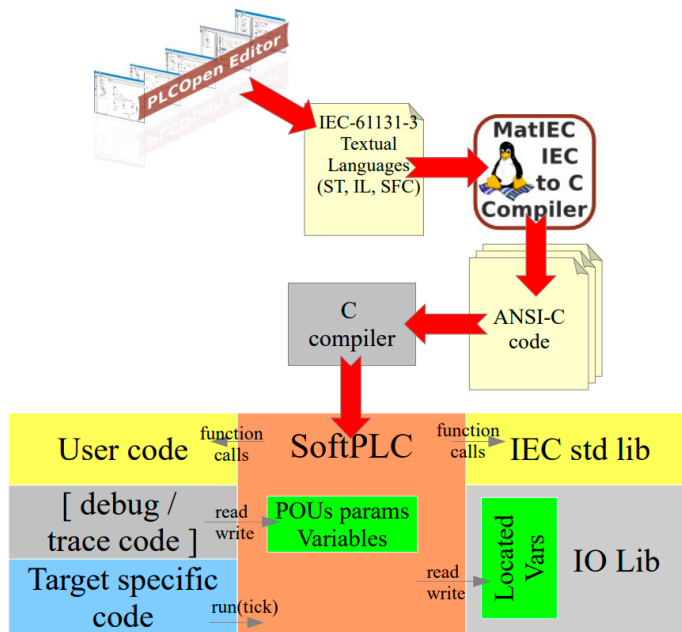


Abbildung 0.2 Übersicht Openplc

Das System besteht aus einer Runtime, das ist die Software, die auf dem Raspberry Pi installiert wird. Diese führt das SPS-Programm aus. Dem Programm Editor, dieser wird auf dem PC unter Windows oder Linux installiert, um das SPS (PLC) Programm nach IEC 61131-3 Standard zu schreiben.

Folgende SPS-Sprachen werden unterstützt:

Programmiersprache	Abk.	Englische Bezeichnung	Abk.
Kontaktplan	KOP	Ladder Logic	LD
Funktionsbausteinsprache	FUP (FBS)	Function Block	FBD
Anweisungsliste	AWL	Instruction List	IL
Strukturierter Text	ST (SCL)	Structured Text	ST
Ablaufsprache	AS	Sequential Function Chart	SFC

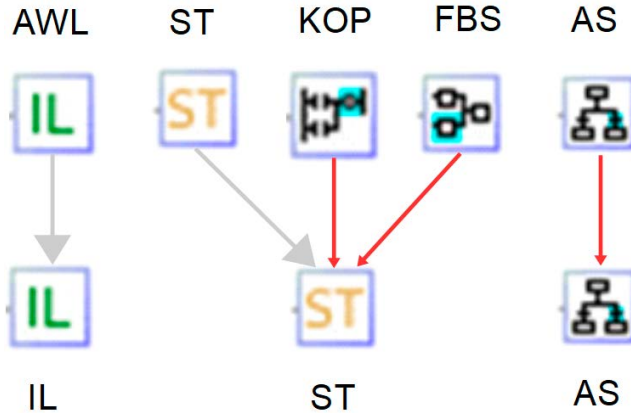


Abbildung 0.3 IEC Programmiersprachen

Die grafischen Sprachen KOP und FUP werden vom Matiec Compiler nach ST übersetzt.

Wir beginnen in Kapitel 1 mit der Installation der Runtime auf den Raspberry Pi. Nach dem Download des Installationsprogrammes erstellen wir eine Micro-SD Karte mit dem Betriebssystem. Nach der Inbetriebnahme installieren wir die SPS Runtime und nehmen einen ersten Test vor.

Danach werden wir uns in Kapitel 2 mit dem Editor und dessen Benutzeroberfläche beschäftigen. Wir laden uns ein fertiges Beispiel herunter und übersetzen es in ein Programm, das wir zum Raspberry Pi übertragen können.

In Kapitel 3 geht es ans Programmieren mit dem SPS-Editor vom „openplcproject“. Wir erstellen eigene Programme in den verschiedenen Programmiersprachen, übersetzen diese und laden sie zum Raspberry Pi hoch, um diese auf der Hardware zu testen

Auch die Visualisierung sollte nicht zu kurz kommen, deshalb sehen wir uns in Kapitel 4 das AdvancedHMI Projekt an. Damit haben wir die Möglichkeit Prozesse, die auf der SPS ablaufen, über Modbus am PC zu visualisieren.

Im Kapitel 5 sehen wir uns an, welche Möglichkeiten es gibt, mit externen Modulen zu kommunizieren. Wir benutzen dabei das weitverbreitete Protokoll Modbus/RTU für den Arduino Uno und das Modbus/TCP Protokoll für den ESP8266 über WLAN. Dabei stellen wir auch Schaltungen und Layouts für diese Hardware vor.

Alle Programmbeispiele können von der Website des Autors heruntergeladen werden. Die Links zu den Webseiten befinden sich im Anhang unter „Weblinks“.

Kapitel 1 • Installation des Raspberry Pi 4

1.1 Hardwarebeschreibung

Der Raspberry Pi ist inzwischen ein weitverbreiteter Mini Rechner, der wegen seines niedrigen Preises nicht nur von Hobby Elektronikern, sondern auch von vielen Firmen in der Industrie eingesetzt wird.

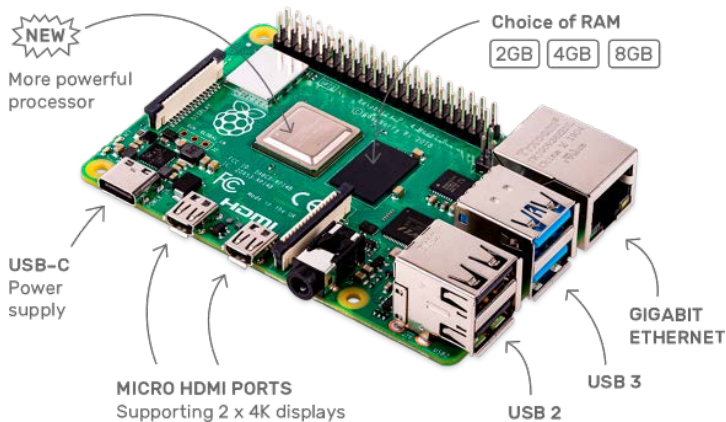


Abbildung 1.1 Raspberry Pi 4

Er besitzt ausreichend Schnittstellen, um als SPS benutzt zu werden. Neben Standard-Schnittstellen wie HDMI, USB, Ethernet, Audio besitzt er eine 40 polige GPIO-Steckerleiste, um mit der Außenwelt in Verbindung zu treten. An diesem Stecker können über passende Interfaces Relais, Taster, Schalter usw. angeschlossen werden. Weiter wollen wir zunächst nicht auf die Hardware eingehen.

Auf der Raspberry Website gibt es mehrere ausführliche Tutorials zu diesem Minirechner.

Wer nach einer kompakteren Lösung sucht, kann auch den Raspberry Pi Zero W einsetzen. Nach einem ersten Test hat alles bestens funktioniert. Wer die Installation von mehreren Stunden in Kauf nimmt, bekommt eine preisgünstige Alternative zum Raspberry Pi 4.



Abbildung 1.2 Raspberry Pi Zero W

Um den Raspberry Pi als SPS einzusetzen installieren wir uns zunächst das Betriebssystem, das wir auf der Website vom Raspberry.org finden. Link (Raspberry Pi OS – Raspberry Pi).

1.2 Installation des Betriebssystems

Wir laden uns hier den Raspberry Pi Imager herunter. Das ist eine einfache und schnelle Möglichkeit, das Betriebssystem auf die MicroSD-Karte zu installieren. Auf Youtube gibt es auch ein Video dazu. Link (<https://www.youtube.com/watch?v=J024soVgEeM>)

Nach dem Download Link (https://downloads.raspberrypi.org/imager/imager_1.5.exe) wechseln wir in dieses Verzeichnis und starten den Imager.



Abbildung 1.3 Raspberry Pi Imager

Nach dem Klick auf „Install“ startet die Installation des Imager Programms, mit dem wir unsere MicroSD-Karte installieren.

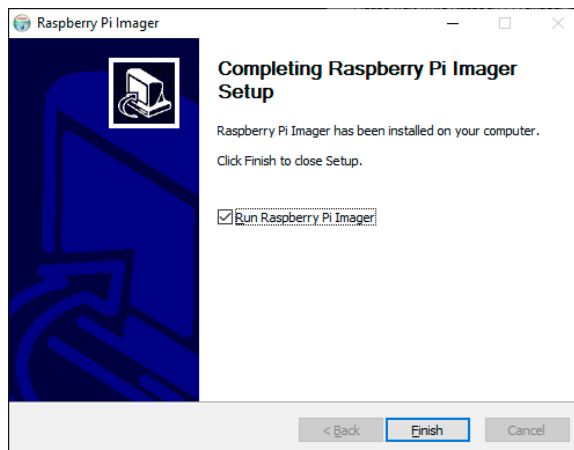


Abbildung 1.4 Raspberry Pi Imager Setup

Mit einem Klick auf die Checkbox „Run Raspberry Pi Imager“ startet dieser nach erfolgreicher Installation.

Wir wählen als Betriebssystem das Raspberry Pi OS (32-Bit) aus. Danach das Laufwerk, an dem die MicroSD-Karte angeschlossen ist, hier Laufwerk G:

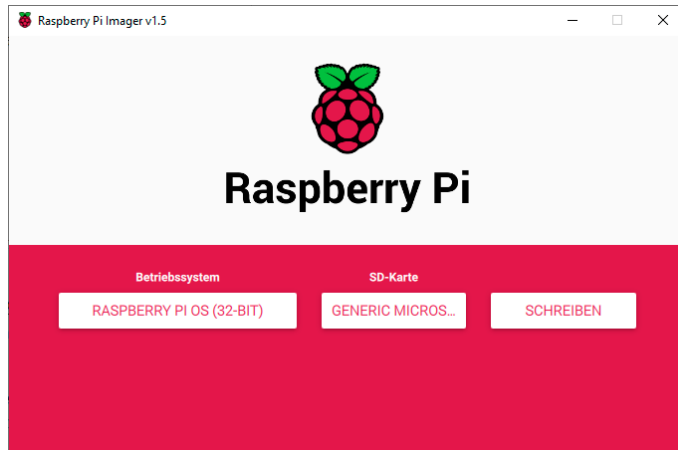


Abbildung 1.5 Raspberry Pi Imager OS Auswahl

Mit einem Klick auf den Button „Schreiben“ starten wir die Installation.



Abbildung 1.6 Raspberry Pi Imager Start

Die Sicherheitsabfrage bestätigen wir mit „JA“.

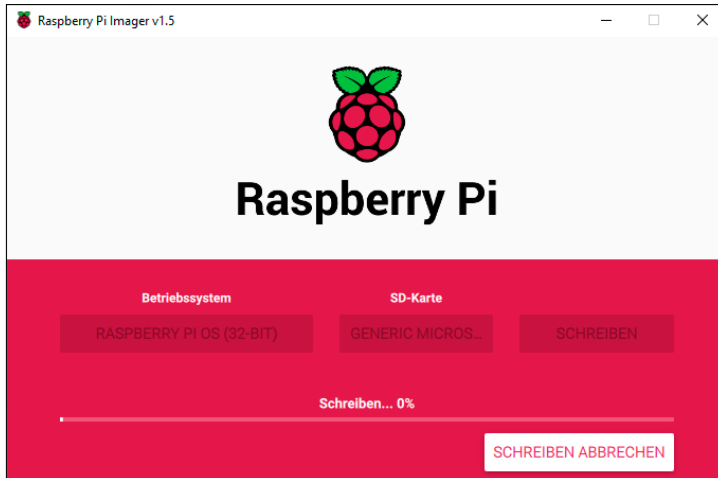


Abbildung 1.7 Raspberry Pi Imager SD Karte schreiben

Jetzt erfolgt der Download des Betriebssystems, und das Schreiben auf die MicroSD-Karte. Das kann je nach Rechner ca. 30 Minuten dauern. Nach dem Hinweis, dass die Karte geschrieben wurde, können wir den Imager mit „WEITER“ beenden.

Dokumentationen zur Installation und zur Benutzung findet man auch hier.

Link:(www.raspberrypi.org/documentation/)

Weiter geht es mit dem Einrichten des Raspberry Pi. Die programmierte MicroSD-Karte in den Raspberry Pi einstecken. Danach für das erste Einrichten des Raspberry Pi einen Monitor, Tastatur und Maus anstecken.

Jetzt verbinden wir das 5-V-Steckernetzteil mit dem Raspberry Pi. Jetzt fährt das Betriebssystem hoch.

Nach dem ersten Start fragt das Betriebssystem nach der Ländereinstellung und der Zeitzone, hier zweimal Deutsch und Berlin eingeben. Jetzt werden die Einstellungen installiert. Danach wird das Passwort verlangt. Standard Benutzername ist „pi“ und das Passwort „raspberrry“

Falls kein Ethernet Anschluss vorhanden ist, richten wir den Netzwerkzugang über WLAN ein. Rechts oben zwischen dem Bluetooth und dem Lautsprechersymbol klicken wir das WLAN Symbol an. Wir wählen unser WLAN Netzwerk aus und geben den Zugangscode ein. Nach einigen Sekunden erfolgt die Verbindung zum hauseigenen WLAN. Jetzt gehts weiter mit der Installation der Updates. Das kann einige Minuten dauern. Danach wird ein Neustart durchgeführt. Beim Anklicken des WLAN Symbols sieht man auch die IP-Adresse. Diese notieren wir uns für die spätere Verwendung mit VNC und der SPS-Software, die wir noch installieren müssen.

Hier: 192.168.178.89.

Später, wenn wir den Raspberry Pi als SPS nutzen, wird es einfacher, über den VNC Viewer darauf Zugriff zu haben.

Über Menü->Einstellungen->Raspberry-Pi-Konfiguration dann unter Schnittstellen kann der VNC Viewer freigegeben werden.

1.3 Installation des VNC Viewers

VNC ist ein grafisches Desktop-Sharing-System, mit dem Sie die Desktop-Schnittstelle eines Computers, hier den Raspberry Pi (VNC Server) von einem anderen Computer oder Mobilgerät mit dem VNC Viewer (Client) fernsteuern können. Der VNC Viewer überträgt die Tastatur und entweder Maus- oder Touch-Ereignisse an den VNC Server und empfängt im Gegenzug Aktualisierungen auf dem Bildschirm.

Diesen können wir schon mal herunterladen und auf dem PC installieren.

Link (www.realvnc.com/en/connect/download/viewer/).

Hier können Sie den geeigneten Viewer herunterladen. Wir benutzen die aktuelle Version für Windows 10. Eine Version für Android ist ebenfalls verfügbar.

Nach dem Download können wir die Installation starten.

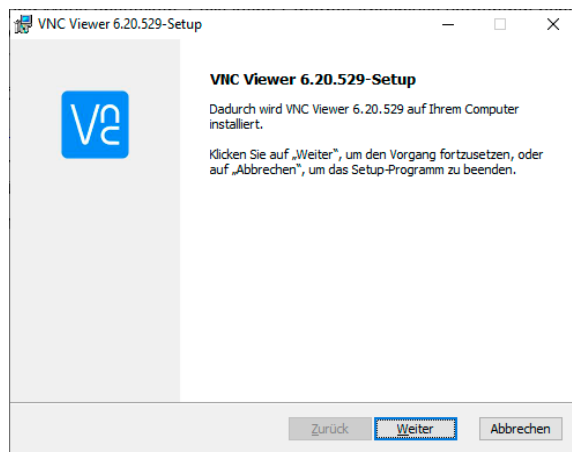


Abbildung 1.8 VNC Viewer Installation

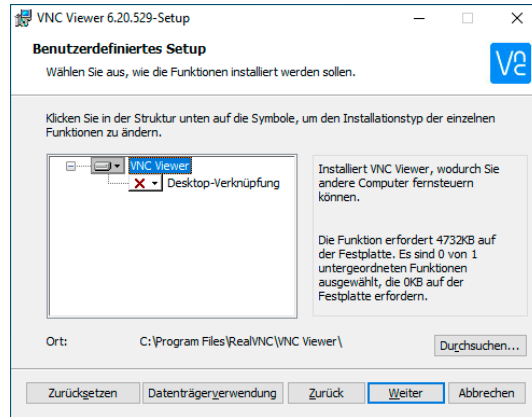


Abbildung 1.9 VNC Viewer Installation

Hier setzen wir die Installation mit „Weiter“ fort. Hier auf „Installieren“ klicken, damit startet die Installation des Viewers.

Jetzt starten wir den VNC Viewer und stellen eine Verbindung zum Raspberry Pi her.



Abbildung 1.10 VNC Viewer Startmenü

Unter „Datei“, „Neue Verbindung“ geben wir die Zugangsdaten von unserem Raspberry Pi ein.

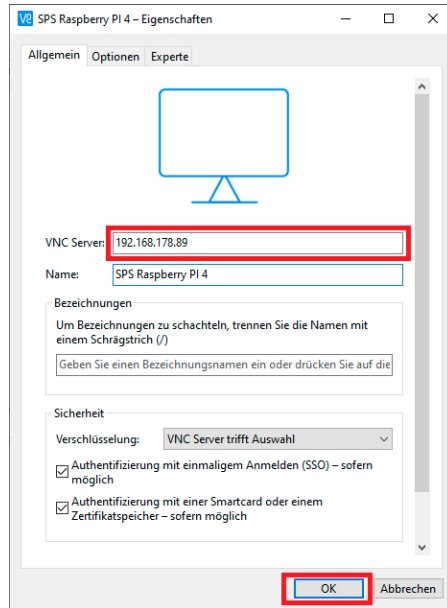


Abbildung 1.11 VNC Viewer Konfiguration

Hier geben wir die IP-Adresse und einen passenden Namen ein. Danach bestätigen wir das „Eigenschaften“ „Allgemein“ Fenster mit OK.

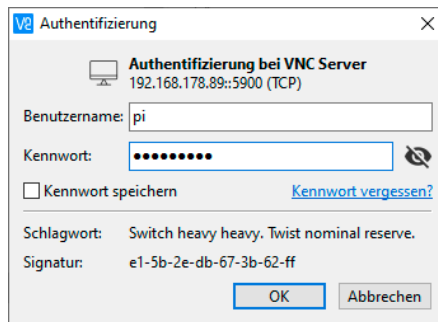


Abbildung 1.12 VNC Viewer Zugangsdaten

Hier geben wir den Benutzernamen „pi“ und das Kennwort „raspberry“ ein und bestätigen mit „OK“.

Kurz darauf erscheint der Desktop des Raspberry Pi auf unserem PC-Bildschirm.

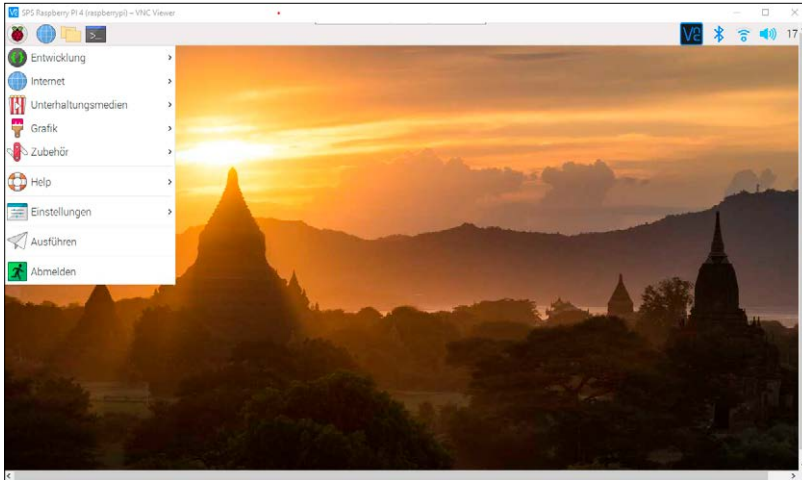


Abbildung 1.13 Raspberry Pi OS Desktop

1.4 Installation der File Transfer Software WinSCP

WinSCP ist ein Open Source SFTP-Client, FTP-Client für Windows. Wir brauchen diesen noch um Dateien zwischen PC und Raspberry Pi zu übertragen.

Um einen Zugriff zu erhalten müssen wir am Raspberry Pi unter Einstellungen → Raspberry-Pi-Konfiguration → Schnittstellen SSH aktivieren.

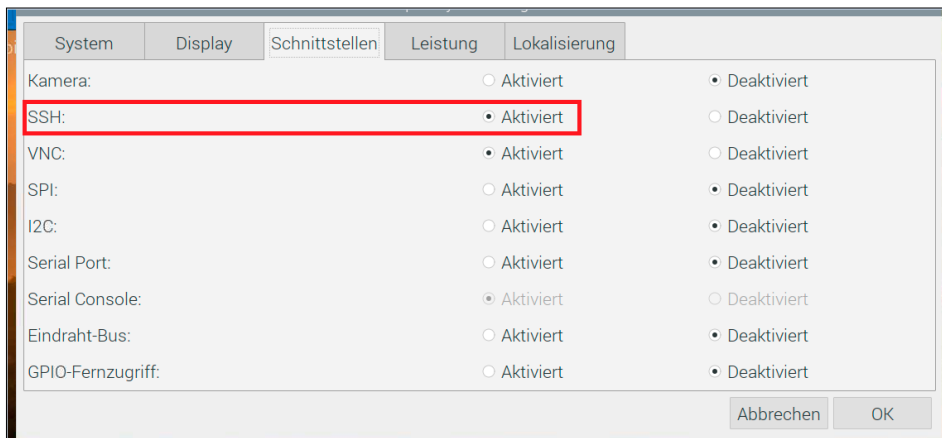


Abbildung 1.14 SSH Konfiguration

Zunächst laden wir das Installationsprogramm herunter und speichern es.



Abbildung 1.15 WinSCP Download

Danach gehen wir in das Download-verzeichnis und beginnen mit der Installation.

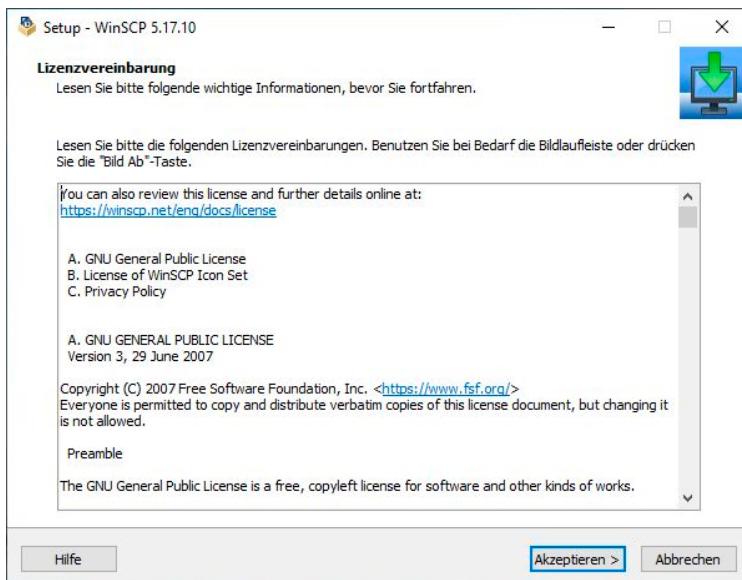


Abbildung 1.16 WinSCP Installation

Wir akzeptieren die Lizenzvereinbarung mit „Akzeptieren“.

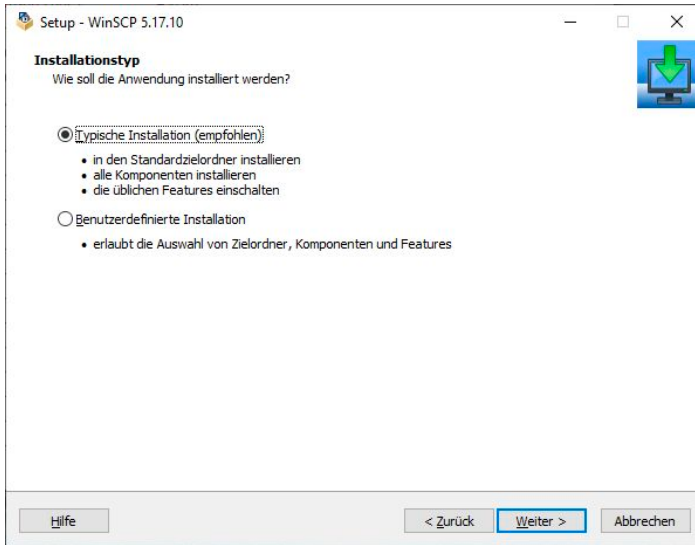


Abbildung 1.17 WinSCP Installation

Hier bestätigen wir mit „Weiter“.

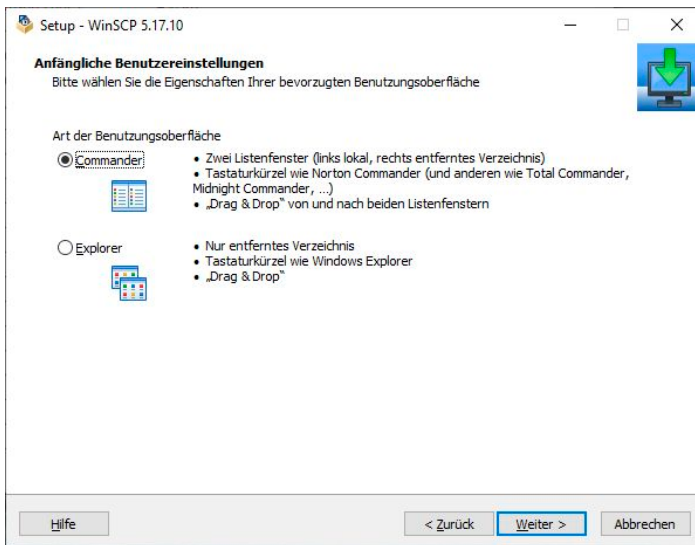


Abbildung 1.18 WinSCP Installation

Wir wählen die Benutzeroberfläche „Commander“, gehen auf weiter und bestätigen mit „Installieren“.

Danach starten wir WinSCP und richten mit der IP-Adresse des Raspberry Pi eine neue Verbindung ein. Hier die IP-Adresse 192.168.178.89 mit den Zugangsdaten.